Lesson 2, Part A: SQL Basics

December 27, 2016

Audience

- Those who are new to relational DBs and SQL.

Objective

To understand basic ***SQL statements***.

To create simple ***Queries: SELECT, DISTINCT, Sorting & Ordering***.

To go beyond simple columns: ***Concatenation, Mathematical, Dates, Null***.

To understand how to ***filter data using the WHERE clause***.

Software & Resources Needed

- Database Server with Interface
  - MS SQL Server with SQL Server Management Studio

  OR

  - MySQL with MySQL Workbench (part of MySQL install)

  OR

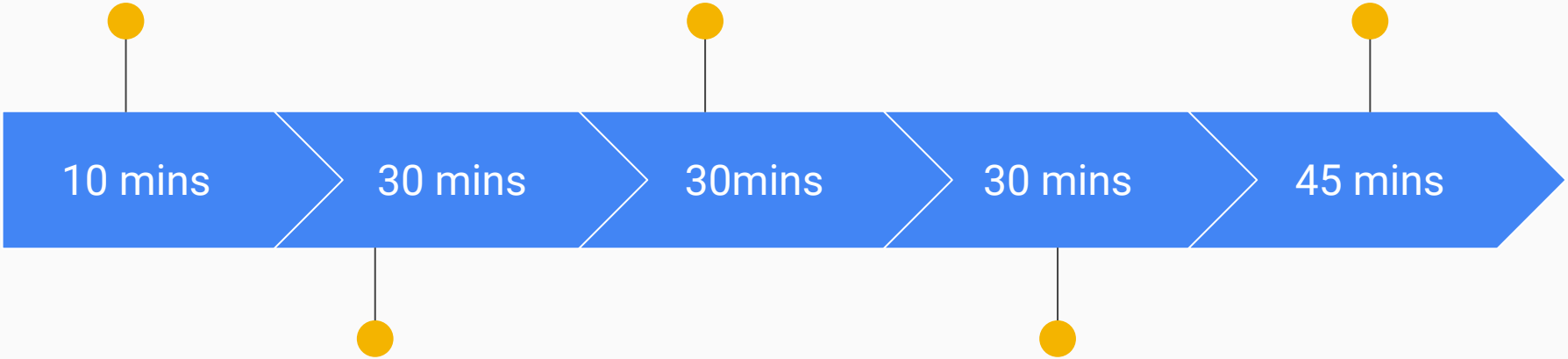
  - Online (sqlfiddle.com)
- Data
  - Test data on Github repo.
- Github repo for information

Overview of Day 2

Basic SQL Queries (SELECT)

Expressions, Casting, NULL

Practice



SELECT, DISTINCT ORDER BY

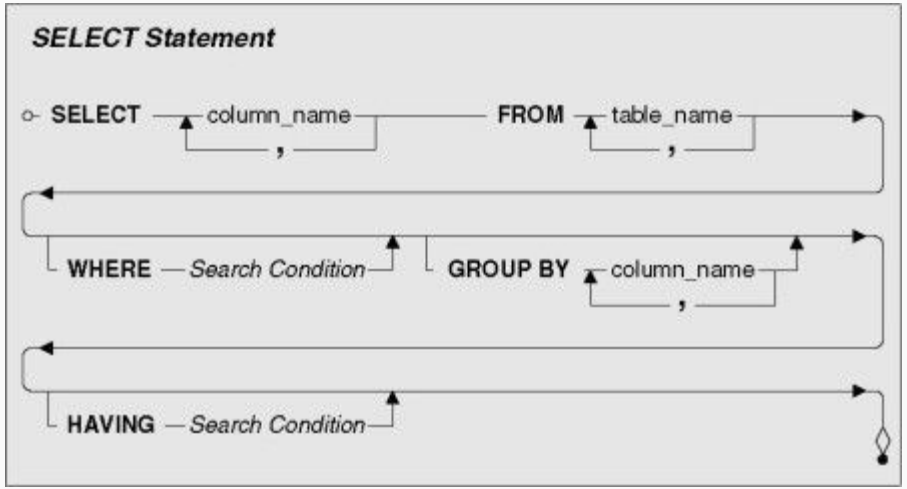
Filtering using WHERE Clauses

Day 2, Part A: Creating a simple query

Basic SQL Queries

Basics of SELECT Statements

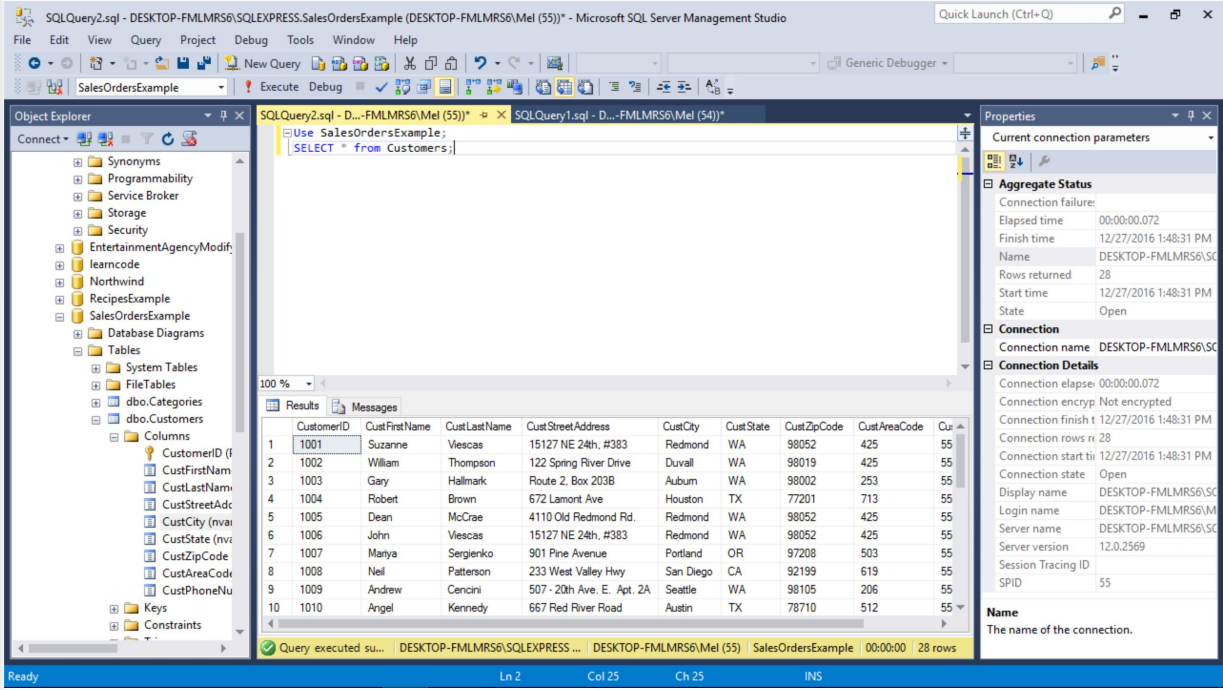
- At the heart of SQL is the SELECT statement.
- You select one or more columns from one or more tables.
- You can choose to filter the results using WHERE or HAVING clauses.
- You can choose to GROUP BY, and ORDER BY (not shown) different criteria.
- **This is how you change mere data into potentially useful information.**
- **See** https://youtu.be/pfsSt2somfg



Example

Open up SSMS or MySQL Workbench and using the SalesOrdersExample database, select all columns of the Customers table.

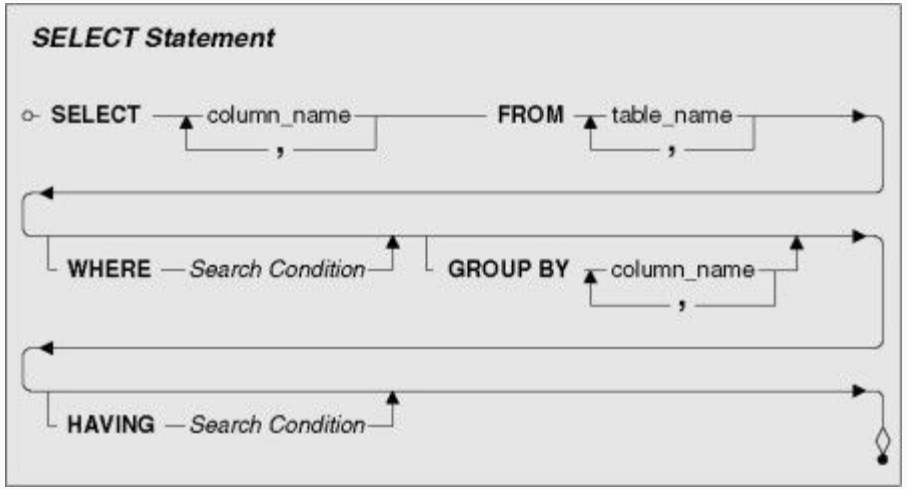
You can select all columns of a table by using an asterisk (*) instead of column names.



Simple Queries: SELECT, DISTINCT, Sorting and Ordering

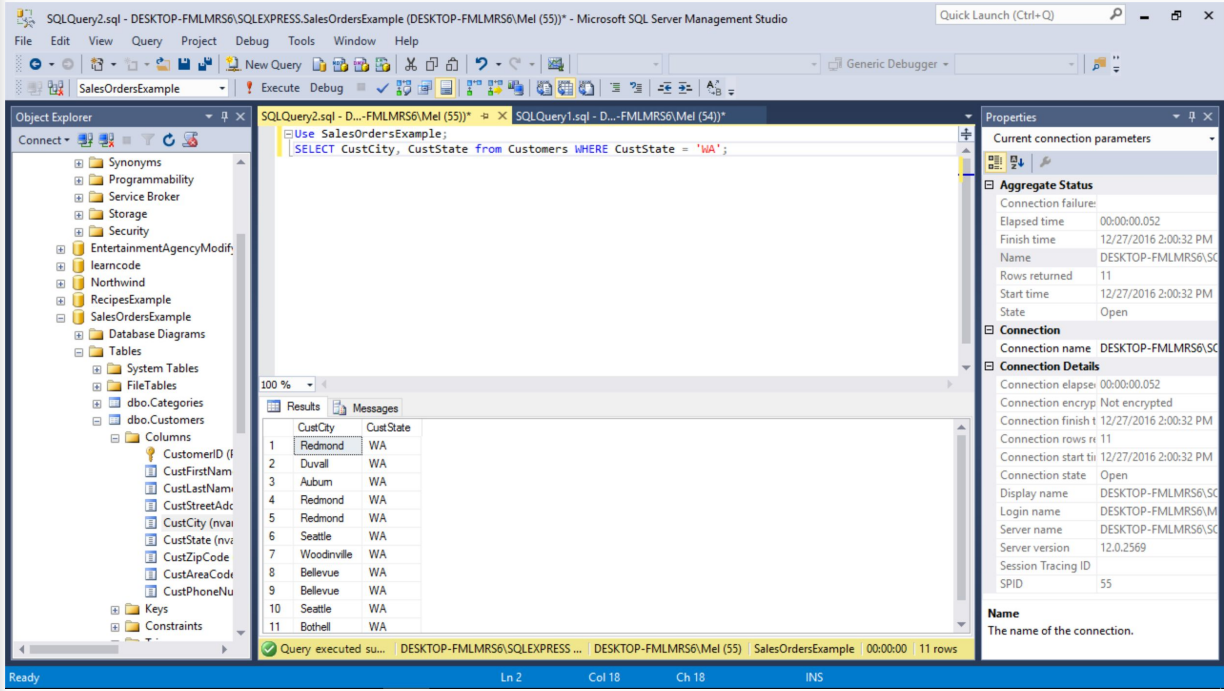
SELECT Statement

- When you query a database, you are using a SELECT statement.
- Formed using clauses (distinct keywords).



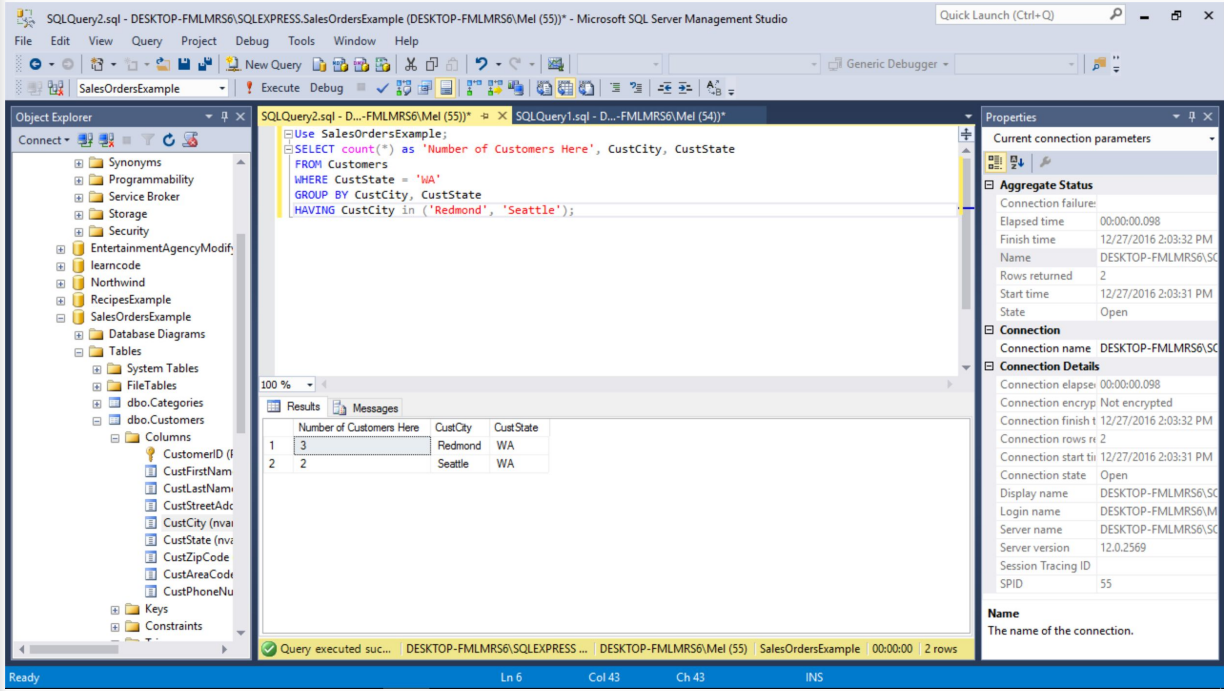
Clauses

- SELECT
  - Required. The biggie. You must include this. Tells DB what information you want to retrieve.
- FROM
  - Also required. Tells db from where to gather the data. You list tables and views here.
- WHERE
  - Optional. Filters the rows returned by the FROM clause.



Clauses (cont)

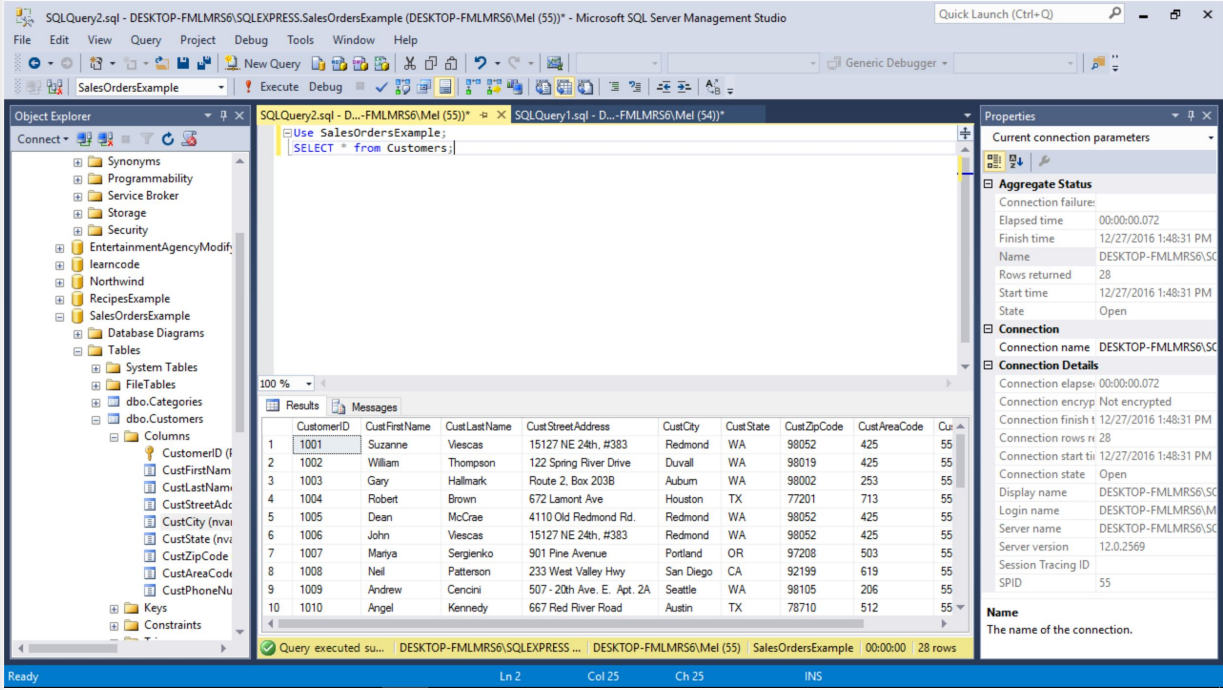
- GROUP BY
  - If you use an aggregate function in the SELECT clause, you can use this clause to section the information into groups.
  - Aggregate functions are built in functions like SUM, COUNT, etc.
- HAVING
  - Similar to a WHERE clause, but used to filter aggregate functions.



Example: SELECT all FROM Table

Open up SSMS or MySQL Workbench and using the SalesOrdersExample database, select all columns of the Customers table.

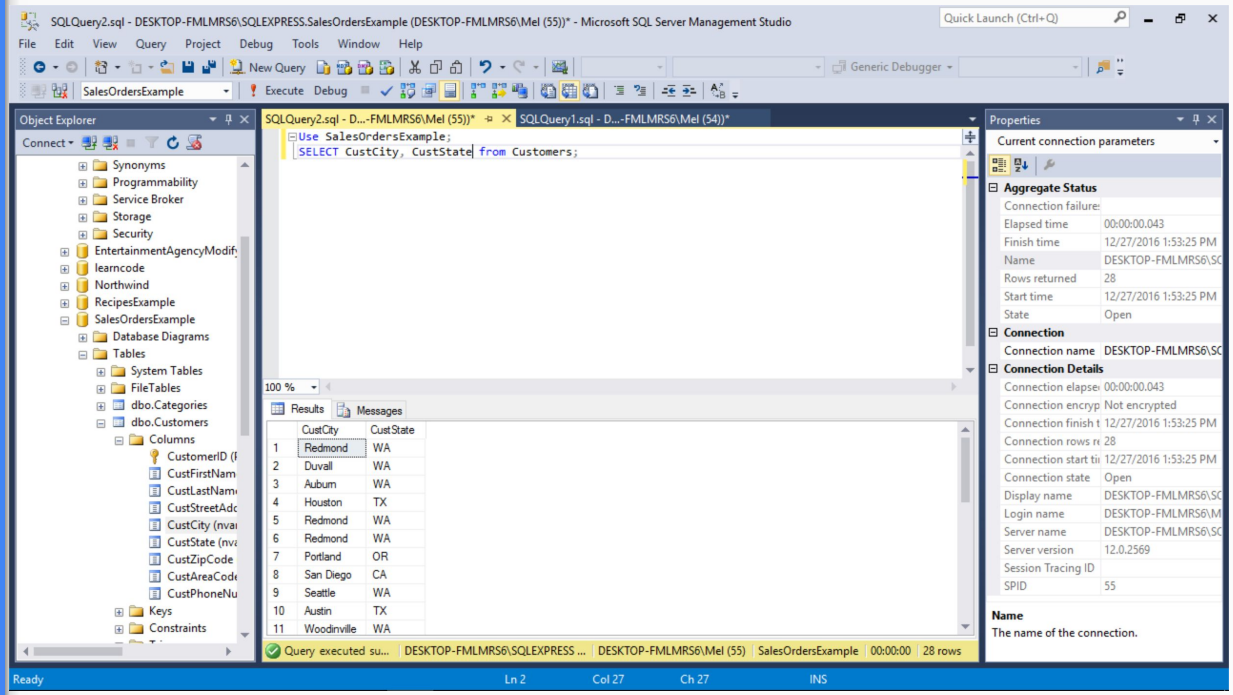
You can select all columns of a table by using an asterisk (*) instead of column names.



Example: SELECT 2 columns FROM table

Continuing with the SalesOrdersExample database, select the city (CustCity) and state (CustState) of all the customers in the Customers table.

You can select multiple columns from a table by listing the column names after SELECT, separated by commas.

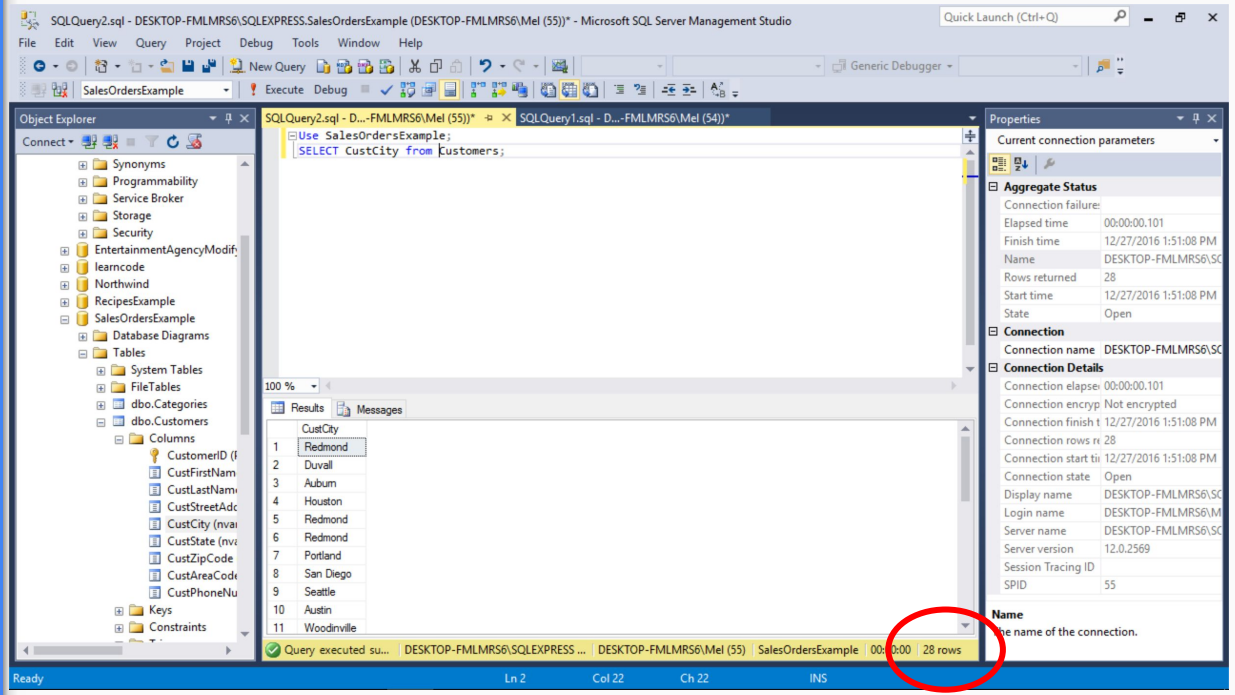


Example: SELECT one column FROM table

Continuing with the SalesOrdersExample database, select just the city (CustCity) of all the customers in the Customers table.

You can select a single column from a table by listing that column name after the SELECT keyword.

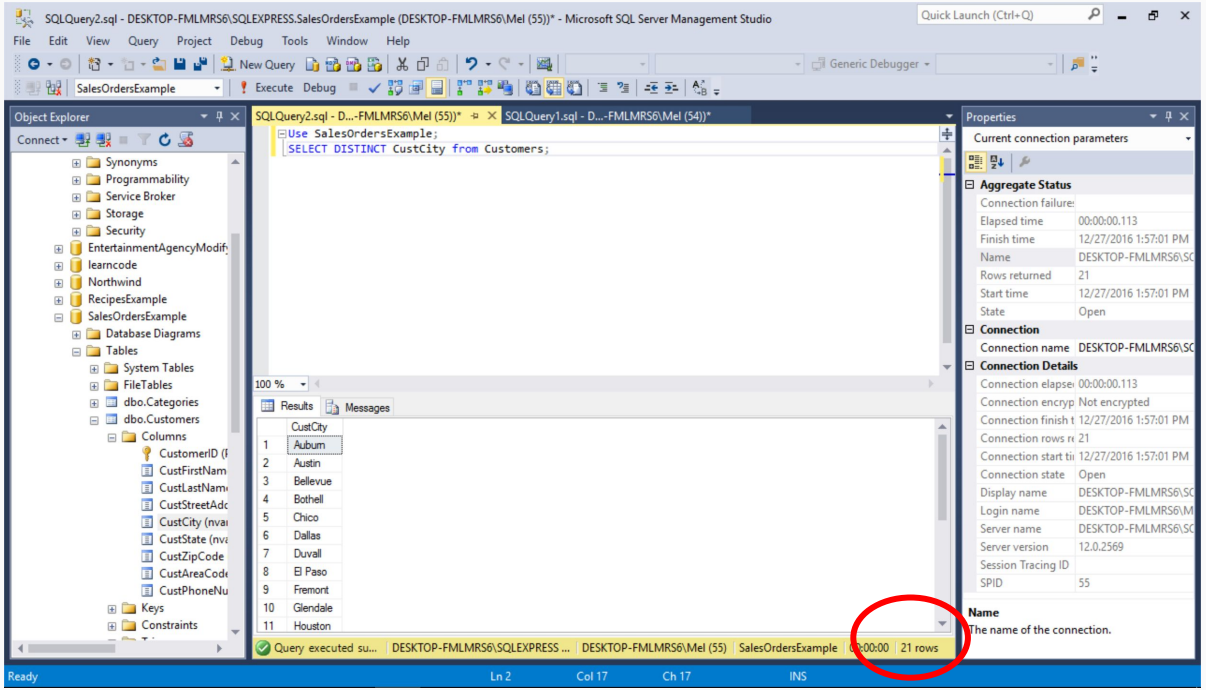
Note the number of rows returned (circled in image to the right).



Example: SELECT DISTINCT records FROM table

Continuing with the SalesOrdersExample database, let's learn how to select non-duplicated, aka DISTINCT, cities (CustCity) from the customers in the Customers table.

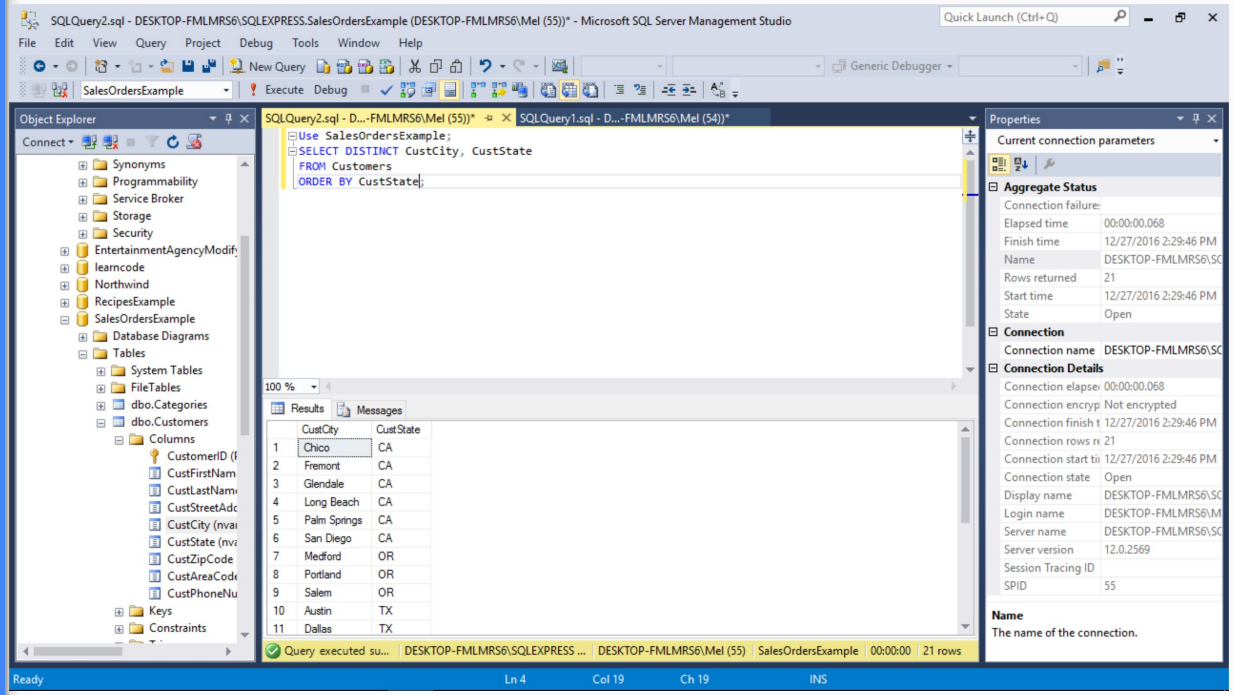
Note: by using the DISTINCT keyword, we don't get repeated results. Now only 21 (see red circle to the right) records are returned instead of 28. This means we have several customers from the same city.



Example: SELECT DISTINCT records FROM table and ORDER BY state

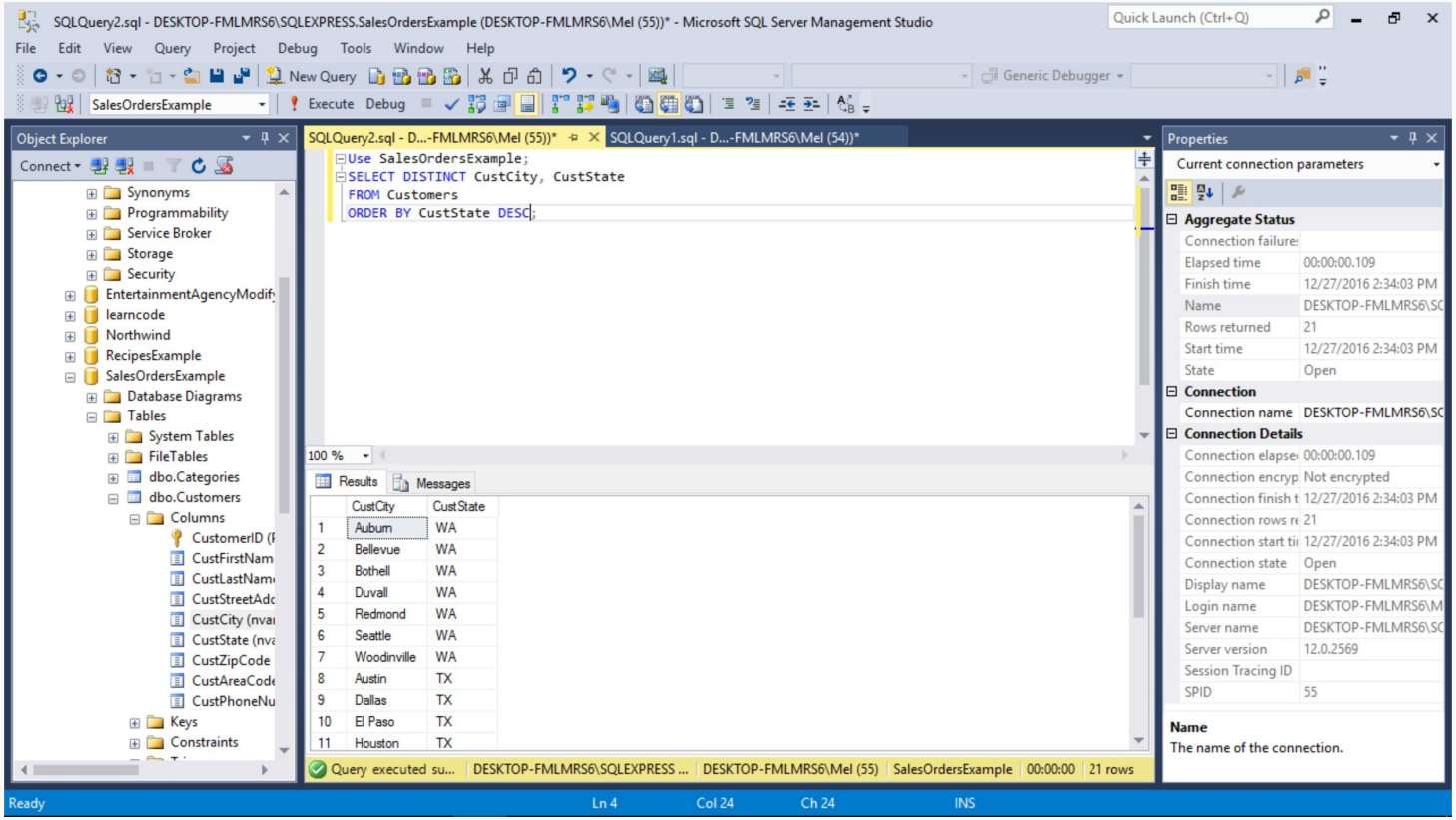
Continuing with the SalesOrdersExample database, let's learn how to select non-duplicated, aka DISTINCT, cities (CustCity) with their states (CustState) from the customers in the Customers table, Ordering the results by state.

NOTE: The result is often ascending by default. Learn how to change to order of the sort in the next slide.



Example: SELECT DISTINCT records FROM table and ORDER BY state DESC

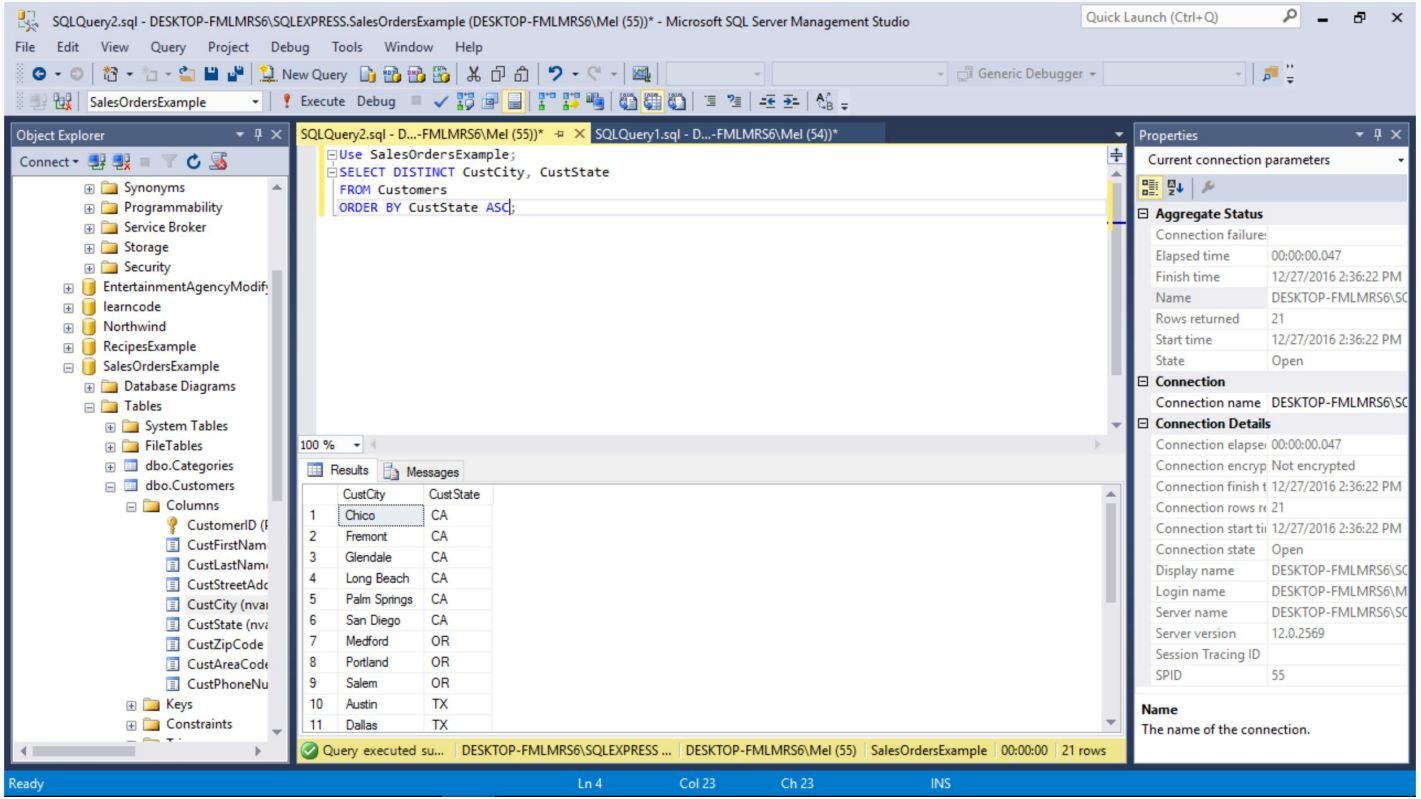
Descending order



Example: SELECT DISTINCT records FROM table and ORDER BY state ASC

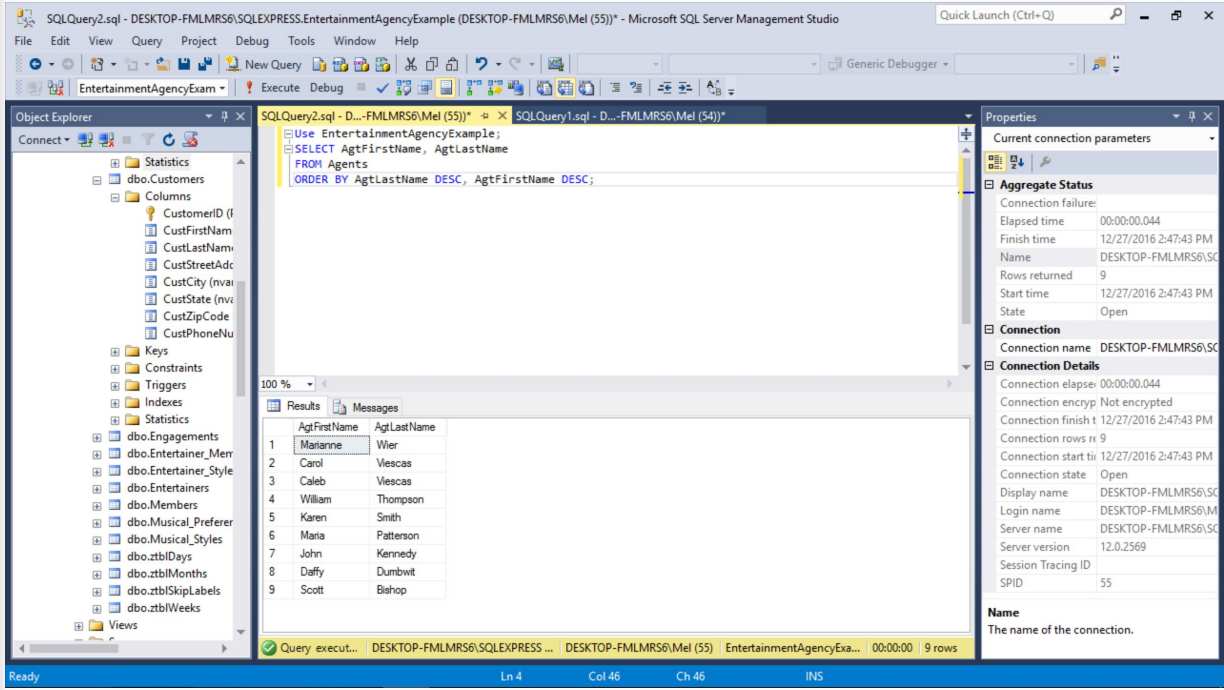
Explicitly stating

Ascending order



Example: SELECT records FROM table and ORDER BY 2 columns descending

Using the EntertainmentAgencyExample database, let's see how to select first and last names of agents, and have the results come to us by last name (descending), then first name (descending).



Try It

Select all records from Agents table in the EntertainmentAgencyExample database.

Try It

Select first and last names of the Customers in the EntertainmentAgencyExample database.

Try It

Select first and last names of the Agents, ordering by last name (ascending) in the EntertainmentAgencyExample database.

Try It

Select distinct states in the Agents table, ordering by state (decending) using the EntertainmentAgencyExample database.

Try It

using the EntertainmentAgencyExample database,
select cities and states from the Agents table, ordering first by state (decending), then by city (descending).

Continued in Day 2, Part B: Getting more than simple columns